Thinking in progress

Journal Item

From these, we were able to identify five basic contexts for the use of IWB.

- Teacher as demonstrator
- Teacher as modeller
- Teacher in control - inviting the pupils (shared)
- Pupils in control with the "teacher" advising (guided)
- Pupils working independently

Demonstration was used to describe teacher input that simply demonstrated the mathematical process e.g. how to add two 3 digit numbers, or software features. Modelling suggests the modelling of mathematical thinking with the aid of the IWB software (metacognitive modelling). Sharing occurs when the teacher has control of the IWB but invites pupils to participate in a task. Guided use refers to pupils leading with guidance from the teacher concerning content, direction or technical issues. Pupils can work independently, either in small groups or alone on IWB. The size of the image and the interactivity means small groups can share ideas with greater ease than if they were sitting around a single computer screen.

Implications for the teachers involved in the project

The teachers adopted models for the use of the technology, at least initially, in a way in which they felt most comfortable. Although the IWB was felt to change the flow, content and pace of the lessons, to begin to use it, teachers did not feel that they had to make large shifts in their classroom practice.

Whilst the teachers recognised the first two teaching contexts, demonstrating and modelling, as part of their general classroom practice they could offer little anecdotal evidence of pupils participating in any form beyond the "come up and show us" model in the whole class activities at the start and end of the numeracy lesson. The identification of the lack of pupil involvement led to discussion about teachers' everyday IWB practice. As a consequence, teachers considered the need to expand the sphere of influence of the IWB into parts of the lesson where they were not taking the lead.

Certain software encouraged the adoption of particular approaches to the use of the IWB by the teachers. The majority of the software being used was for whole class- teacher centred work. Where the software had the potential for pupils working independently, it required additional impetus for change. Even when the software had the potential to support pupils working independently this facility was not utilised. The following examples in Figure 1 are those that the teachers came up with in the session. They are not necessarily presented as the best practice in these areas.

Although the purpose of this article has not been to discuss the role of alternative means of achieving similar practices with other technologies – it does appear to be the case that the "interactiveness" of the IWB comes more into its own as you move "down" the contexts that appear in Figure 1. That is, models of teaching and learning that are more open and social are more likely to result in the IWB offering a unique feature to the classroom.

By identifying a framework for "the use of the interactive whiteboard" the teachers were alerted to a wider variety of teaching and learning contexts and possibilities. This has begun to influence their practice and enabled them to engage their established understanding of the flow of pupil/teacher activity in a lesson with their developing use of the IWB. The IWB thus becomes an instrument of change as it becomes more fully adopted and teachers adopt new models of classroom practice that more fully utilise the potential of the whiteboard.

The mapping of a possible framework for the use of the interactive whiteboard meant that teachers now had a way to refine their research questions so that the reader and writer could have a greater chance of having the same understanding as to how the IWB was being used, rather than relying on assumptions as discussed in our introduction. The need for such a framework is clearly evident, the one presented here is intended to stimulate debate as to the precise nature of such a framework.

Penny Knight and Jennie Pennant work with the Bracknell Forest LEA and Jennifer Piggott works at the Faculty of Education,University of Cambridge.

Reference
BECTA,2004,Getting the most from your interactive whiteboard;a guide for Primary schools.

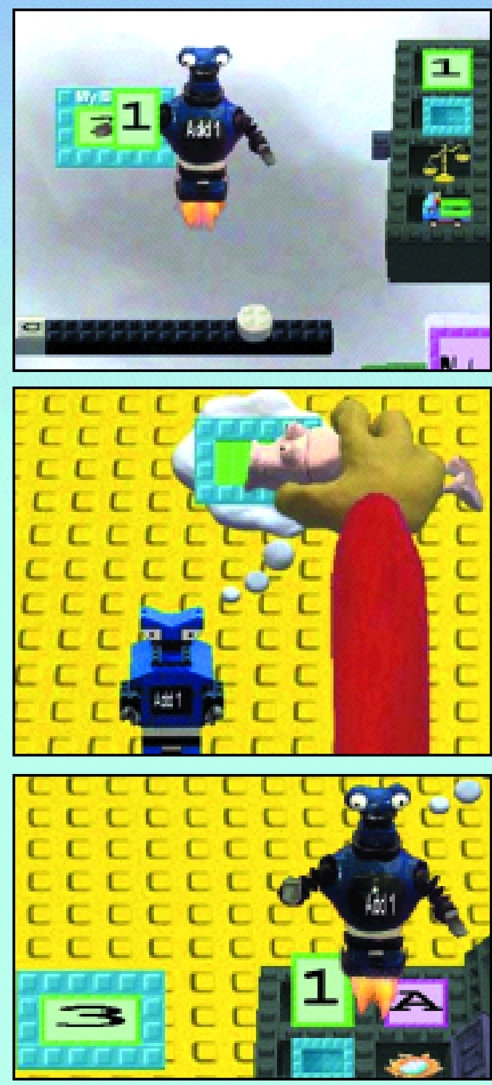
Note:Easiteach Maths is an RM product used in many Primary Schools.The latest version is now part of the Easiteach Studio package. More information can be found on the RM website http://www.rm.com/rmcomhome.asp

Thinking in progress

Yishay Mor, Celia Hoyles, Ken Kahn, Richard Noss and Gordon Simpson

This article tells the story of 11-14 year-old students using a rather new and relatively untested programming system to represent and discuss some deep mathematical ideas. There is one caveat: we have chosen the episodes to indicate the possibilities that can emerge when children are given new ways to talk and think about mathematics. We cannot lay claim to any generality: we would invite the reader to look where we are pointing, not at our fingers!

We are programming with *ToonTalk*, a language we have used in the past with younger children to construct video games, and which we have written about previously in Micromath. *ToonTalk* is a computer game, programming environment and programming language in one. In most languages, programming means writing text (code) in a highly structured syntax. Some environments allow the user to replace pieces of this text with iconic representations. But *ToonTalk* takes this one stage further, in that programs take the form of animated cartoon robots (*ToonTalk* is so named because one is "talking" in (car)toons). Programming is done by training these robots, leading them through the task they are meant to perform. After training, programs are generalised by "erasing" superfluous detail from robots' "minds" (to train a robot, you just go in and program an example in its thought bubble!). So the process of what it means to program is very



Train the robot to take a number 1 from the toolbox and drop it on the input, to increment it.

Generalise the program by erasing the value of the input from the robots memory.

Give the robot its input box. The robot will continuously repeat the actions it has been taught.

Figure 1: Training a robot to count

different from any existing programming language, and because programming objects are animated, it is very difficult to conjure up the feel of the process without trying it. Fortunately, this is straightforward. The important point is that the process is made concrete in a robot, which can be pointed at, named, picked up, and moved about.

Figure 1 shows three snapshots of what it means to write a program (train a robot) to count through the natural numbers. In fact, we only have to train the robot to "add 1" to a number and then *generalise* it to any number. The robot iterates the actions it was trained to do, for as long as the conditions it expects hold.

Alongside the programming system, and a set of appropriate tools developed within it, we have constructed a web-based system we call *WebReports*, designed to allow students to share and discuss the models of mathematical objects and processes they have built in *ToonTalk*. The students use a visual on-line editor to compose reports detailing their work, can comment on and annotate each others' reports, and – most importantly – can publish *working ToonTalk models* of their ideas as they develop. This is a simple process: students can grab any object in their *ToonTalk* environment, and include it in their on-line report. The object is shown as an image, but it is also a hyperlink, which, when clicked, causes the object to open in the *ToonTalk* environment – which could be in another classroom or another country. This last point is crucial: rather than simply discussing what each other *thinks*, students can share what they have *built*, and *rebuild* each others' attempts to model any given task or object. Figure 2 shows the welcome page of a "WebReport", seen after logging into the system. It includes a personal navigation sidebar, tabs to access general areas (such as topics and groups), and a quick view of the most recent updates.

After about a year of work, we are beginning to notice how students are starting to use their programmed models as elements of their discourse. By expressing themselves through their models, they are beginning to find ways to talk about deep mathematical ideas without the necessity *first* to become fluent in algebraic symbolism, and are becoming accustomed to posting their developing ideas on the web, commenting on other students' reports, and challenging them to extend their mathematical ideas.

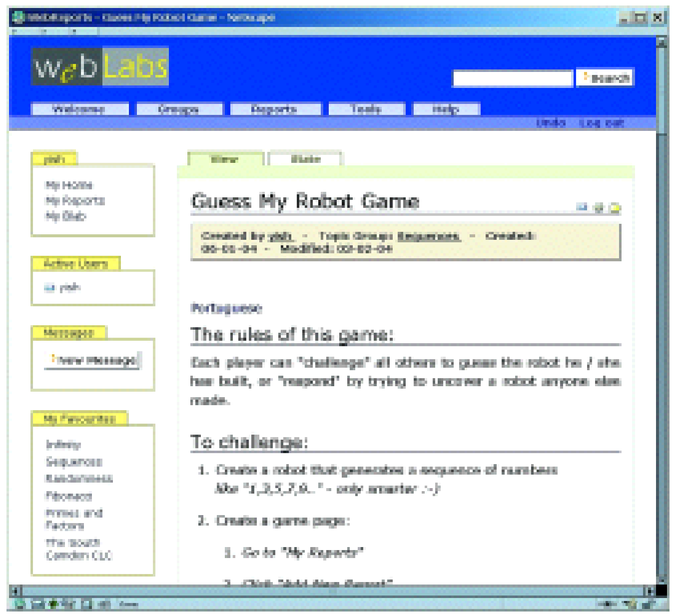
Figure 2: WebReports welcome page

Guess My Robot: Rita's challenge and Nasko's response

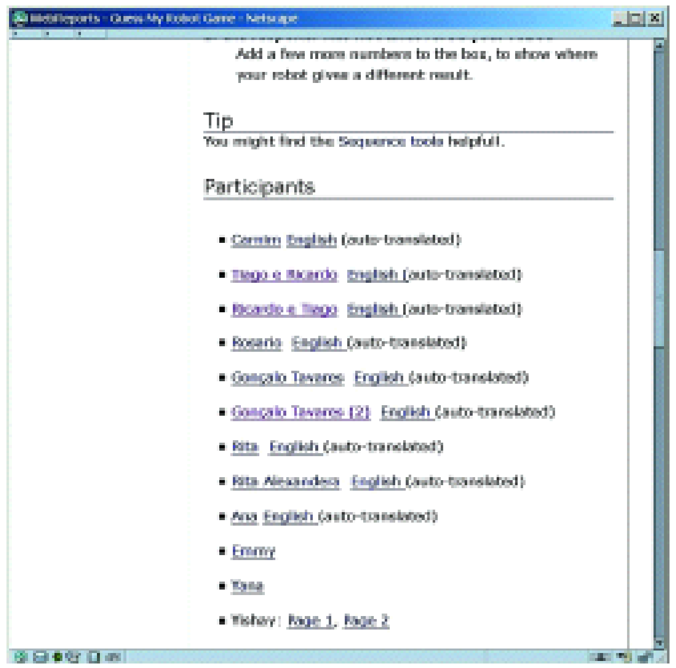
We now turn to the story. First, we should introduce the setting. We focus on the interactions between two groups of students, one in Sofia and the other near Lisbon. The Sofia group consists of 6 boys and girls, aged 11-12, working with *WebLabs* researchers. They have been working with *ToonTalk* for several months, approximately once a week for a couple of hours. The second group is from a village south of Lisbon. Paula, a teacher and researcher in the *WebLabs* team, worked there with a school group (aged 12-13) during the first project year. Researchers in both groups act as teachers, guiding the students through the mathematical ideas as well as through the programming skills. At the same time, the researchers facilitate interactions, by pointing children to interesting peer reports and helping them to add a few words in English to their own reports.

The activity we designed was based on the well-known "Guess my rule" game, employed by us in the context of Logo and spreadsheets, and by many teachers and researchers as a well-known activity. It has also been used in many classrooms in the UK over many years to provoke children to discuss and compare the formulation of rules, and in particular the equivalence (or not) of their algebraic symbolism. In its classical form, it has been used as an introduction to functions and to formal algebraic notation. As Carraher and Earnest (2003) have recently reported, even children in younger grades enjoy participating in this game, and can be drawn into a discussion of algebraic nature through using it. It is an excellent context for students to come to understand that different articulations of *their* constructions can indeed yield the same results and it does this as children feel some ownership of their construction and are willing therefore to engage with others to compare and contrast solutions.

The aims of "Guess my robot" are to encourage students: to build sequences with robots and to challenge others to program the robot that made the sequence; to compare the robots used and to discuss the different methods; and to take a robot



The "Guess my Robot" game presented in a WebReport that explains the game rules.



Each student creates her own game page, and the pages are listed at the bottom of the main game page.

Figure 3: "Guess My Robot" WebReport pages

that "encapsulates the process" and use it to generate new sequences by using new inputs.

So, in our game, *proposers* (students) invent a rule and program it so that it generates a numerical sequence, and publish the first few terms it generates in a *WebReport*. *Responders* then have to build a robot that will produce this sequence, and thus work out the underlying rule. The new element in our variant of the game is that "rules" have to be encoded as robots: one responds to a challenge sequence by posting a robot that produces "the same" sequence. So the encoding of the rule takes the form of a process-description in the form of a ‘program’. Managing to reproduce someone else's sequence by training a robot, is a way to show that you have grasped how the sequence may have been originally generated. As one girl said:

"So, like, the robot is my proof that I got it?"

Let's go back to our story. Figure 3 shows the main "guess my robot" game page. Students enter the game via this page, where they are challenged to generate a "smart sequence" and post it on the web.

Rita is a 14 year old girl from Lisbon, who has been participating in *WebLabs* since February 2003. She likes maths, but has not yet learnt much about sequences in school: this topic is not highly developed in the Portuguese curriculum. In fact, most of her experience in this topic comes from her involvement with *WebLabs.*

Rita found the ‘guess my robot’ activity, and decided to pose her own challenge. The sequence she posted (see Figure 4) was:

2, 16, 72, 296, 1192 …

A few days after she posted it, the Sofia *WebLabs* group held a session, and some of the students tried solving Rita’s challenge. Nasko, a 12 year-old

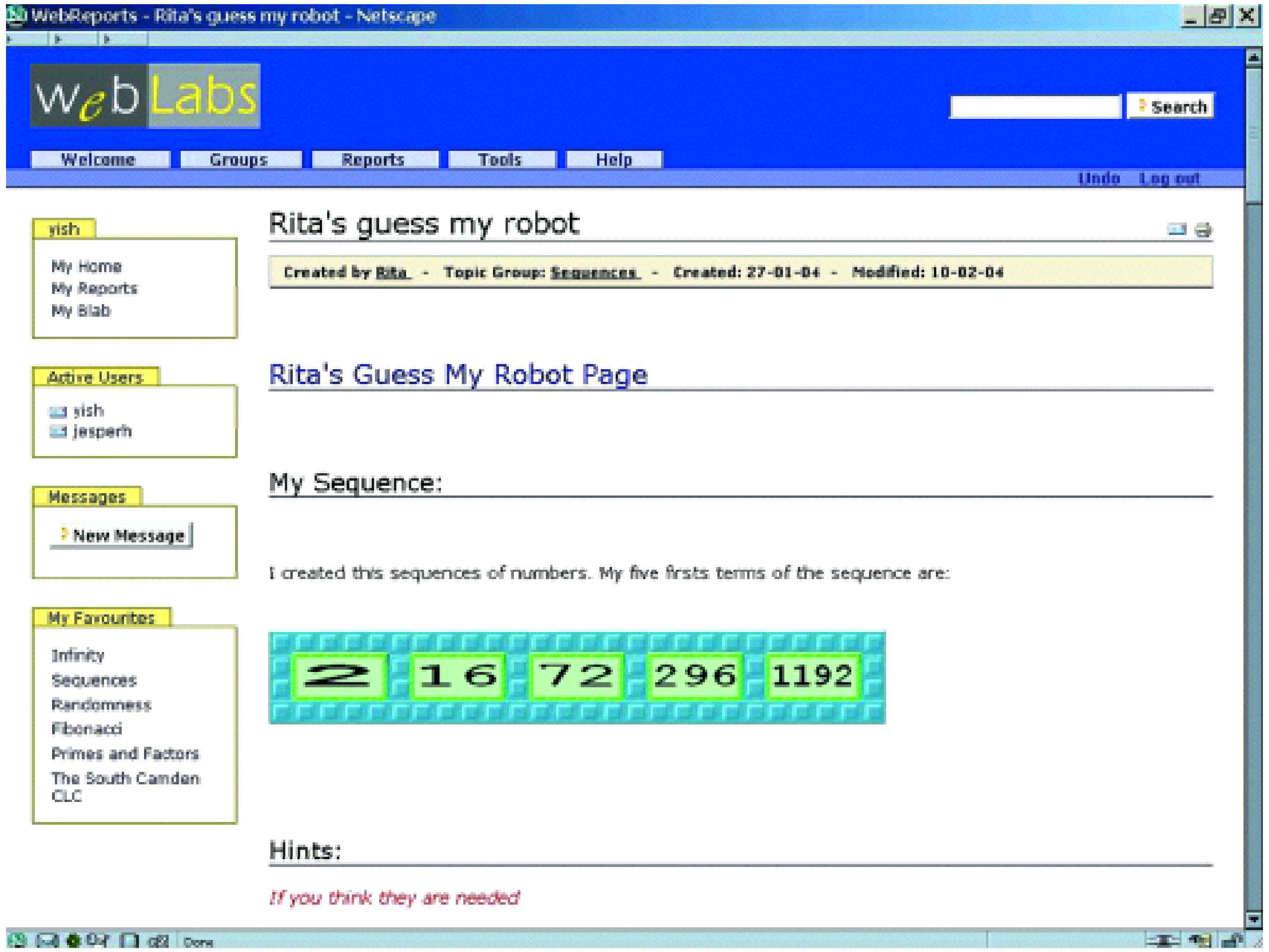


Figure 4: Rita's "Guess My Robot" challenge

boy, posted his response. He had built a robot that produced Rita's five terms, but also realised that the same robot could be used to generate other sequences by changing its initial inputs. So, he posed a two-part challenge for Rita:

- Could she use *his* robot to generate a new sequence of five terms?
- Could she use *her* robot to generate the same sequence?

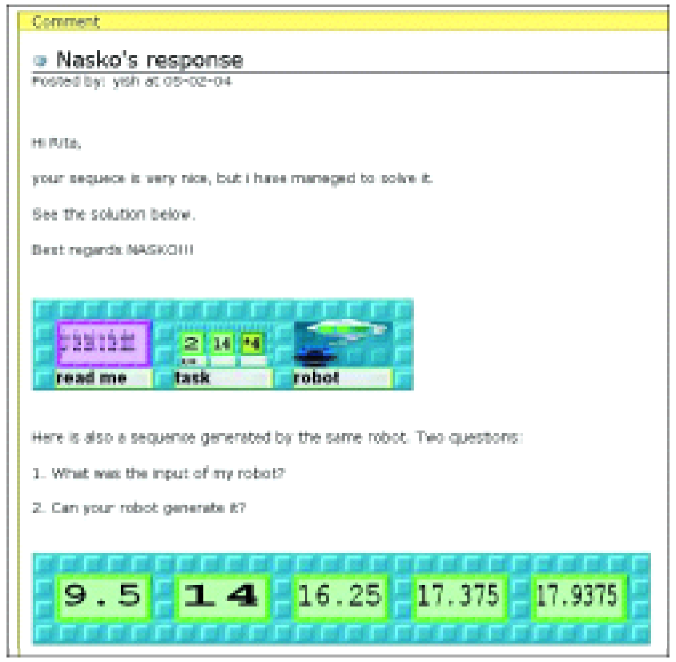


Figure 5: Nasko's response to Rita and his twofold new challenge

Nasko’s response and challenge are shown in Figure 5.

Ivan, who is also a member of the Sofia group, could not participate in that particular *WebLabs* session. Still, that did not stop him from trying his hand at Rita’s challenge, and posting his own response (Figure 6). Ivan succeeded in building a robot, but was especially proud of what he thought was a clever solution as it was ‘simpler’: (*"I only use two holes in the box"*). He adds: *"I’m curious to see other solutions"*.

A few days later Rita came to her next session. She was very excited to find comments on her page – and from children on the other side of Europe! She immediately clicked on the *ToonTalk* robots in the responses, and watched them step through the process of rule-generation. She was totally surprised: Nasko and Ivan had solved her challenge, but their robots seemed completely different from hers (and one from the other)!

Comparing solutions

We will now look briefly at two of the robots (Rita and Nasko’s) and try to clarify, *from an algebraic perspective*, what mathematical ideas are put into play in this story. *ToonTalk* robots operate repeatedly on the data in their boxes. This makes recursive computations very intuitive: just like a spreadsheet, the values in use at step *n* are used to

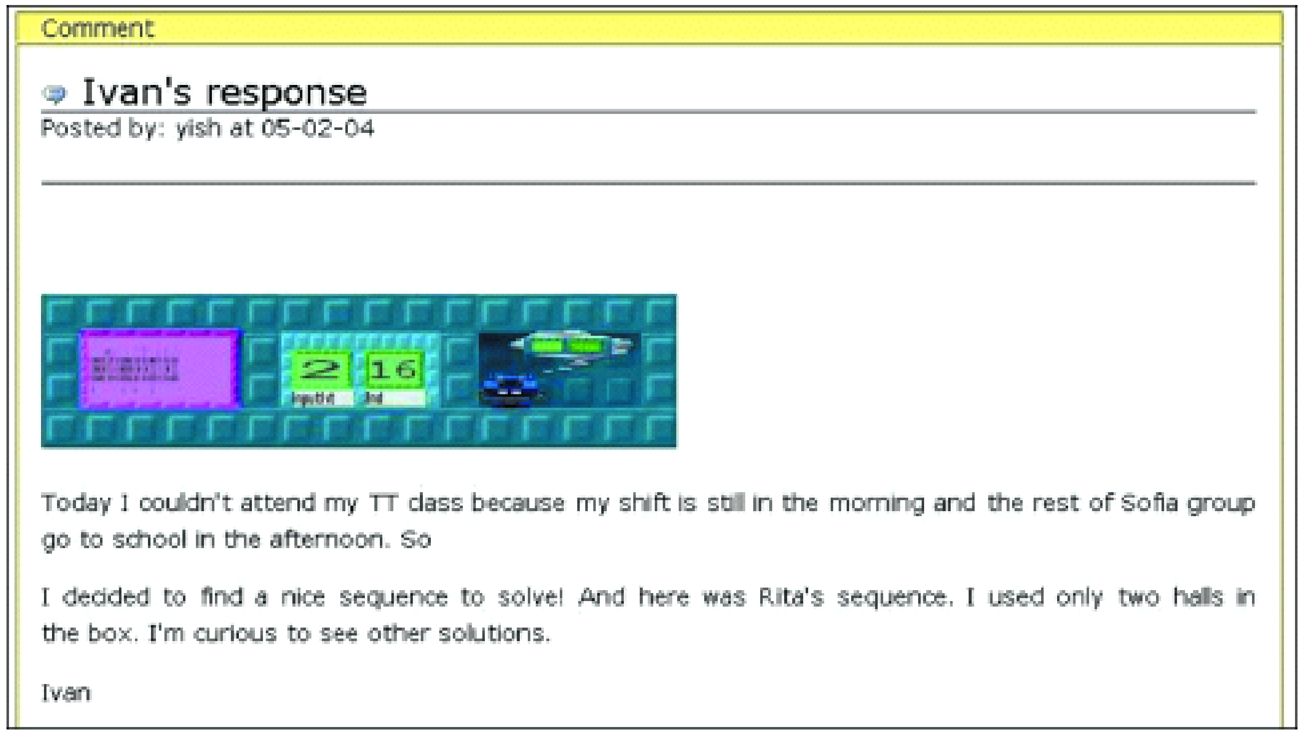


Figure 6: Ivan's response to Rita

Copy the value of A_n to the bird, to send it out, so the first term is the number in the left-most box.

Copy *i* (2) and place on top of A_n, which adds 2 to it.

Copy *k* (*4) on to A_n, multiplying it by 4.

Figure 7: Rita's robot in action

derive the value for step *n+1*. Snapshots of Rita's robot as it runs through its first cycle, are shown in Figure 7.

When this cycle completes, the number in the left-most box is then passed to the "bird", as a second element of the sequence, starting the second cycle. A "bird" is the way that *ToonTalk* uses to communicate between objects. In this case, it takes the number of the sequence in turn, and collects them (on its nest!).

Rita's robot therefore computes:

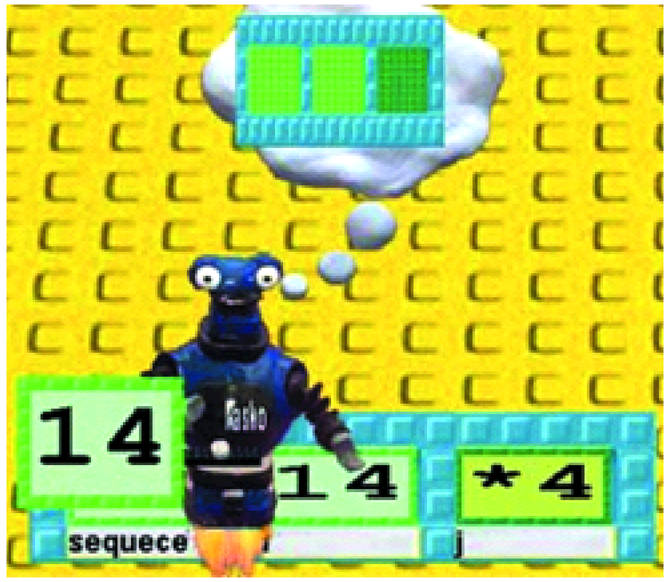
$$A_n+1 = (A_n +2) *4$$
$$A_1 = 2$$

Nasko's robot (Figure 8) differs from Rita's in two respects. Firstly, its programming style is different; while Rita uses a bird to carry the sequence elements out, Nasko displays them in the robot's box. The second disparity is more interesting – the robot appears at first sight to be computing a completely different sequence!

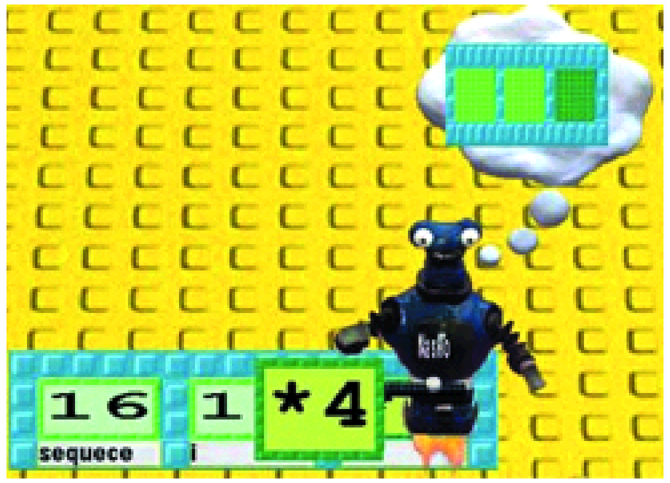
Nasko's robot computes:

$$A_n= A_{n-1}+14) *4^{n-2}$$
$$A_1 = 2$$

Are these two robots equivalent? And what, in any case, does "equivalent" mean (this in itself is an interesting talking point for the two children concerned). This may be an interesting exercise in algebra for the reader.



Copy the value of *i* (14) over the current sequence term (2), adding them up.



Copy the value of *j* (*4) over *i*, multiplying it by 4.

Figure 8: Nasko's robot in action

The next step

Now that something unexpected has happened, the next step is to challenge the students to explain it. Both groups are scheduled to discuss the different solutions, compare them and explain how they appear to generate the same sequence. The questions they will discuss include:

- Explain how you worked out what the sequence was, and how you ended up with the robot you built.
- We think your robots will generate the same sequence forever, but how can we be sure?
- Discuss the different robots, and explain why they seem to generate the same sequence.
- Describe what the robots are doing on the whiteboard or on paper, in a way that will make it easier to compare them.

As at the previous stage, we are planning for surprises. We designed the original challenge in the belief that responders would solve the same task differently from proposers, and that both sides would find this interesting enough to have a *mathematical discussion*. This time, we're prepared to be as surprised as the students. Will they base their explanations on the *ToonTalk* representation, or will they ask for alternative notations? In either case they will be exploring new mathematical terrain. The objects of discussion have emerged from their own activities. This sense of ownership allows, we believe, children to access structures and ideas which may normally be considered too advanced for them. If this is the case, although the *ToonTalk* representation cannot help furnish the proof, it may be a useful tool in motivating an interesting and non-trivial piece of mathematical problem-posing (the situation is somewhat analogous to that of Dynamic Geometry in this respect).

We do not mean to restrict the role of programming to a matter of motivation. We wrote *"the robot computes the sequence…"*. In fact, we should have written *"The robot IS the sequence…"* in the same way as we talk of *"the sequence* $A_n=a+b*n$*"*. The same sequence can be represented as "$A_n = A_{n-1}+b; A_1= a$" or by the robot with a 3-hole box. All three notations are valid, precise and well-defined. Just as a formal equation defines a unique sequence, or class of sequences, so does a *ToonTalk* robot.

As mathematically-educated people, we may take for granted the meaning of a statement like $A_n= a+b*n$. In much the same way, participants in the *WebLabs* community are coming to share the meaning of *ToonTalk* robots as representations of mathematical entities. Each representation has its strengths. While the algebraic form is more conducive to formal manipulations and proof, the *ToonTalk* representation is situated within a familiar activity, and is easier to tweak in an exploratory manner. Most important, what you "write" (i.e. do on the screen) has a real *effect* in terms of what the program produces (how do you know when you've got an algebraic calculation wrong?).

To illustrate this point, we focus on Rita's response to Nasko's challenge. Nasko had challenged Rita to the use the robot he had constructed for her sequence *2, 16, 72, 296, 1192,...* to generate another sequence: *9.5, 14, 16.25, 17.375, 17.9375*

Rita responds to Nasko by comparing each step of the process that the robot acted out, and imagining what numbers the robot must have worked on, in order to obtain the required number:

> *"In Nasko's box for my sequence he used 2 in the first hole, and added 14 from the second hole to get the second term (16), and multiplies that by 4. Then I think to get the first term of Nasko's sequence I need 9.5 in the first hole. The number in the second hole has to be a number that you add to 9.5 to get 14 (the second term), this number is 4.5. In my sequence he uses x4 to get the third term (16 + 14 x 4). Then for his sequence I think like this: 14 + 4.5 "something" should give me 16.25, or 4.5 "something" should give me 2.25. But 2.25 is half of 4.5, then in third hole of the box I need to put /2"*.

Does this count as a mathematical solution? We leave it to the reader to decide.

Yishay Mor , Celia Hoyles, Ken Kahn,Richard Noss and Gordon Simpson are involved in The *WebLabs* Project ,Institute of Education,University of London

Reference

Carraher, D. & Earnest, D. (2003) Guess My Rule Revisited in *Proceedings of 27th International Conference for the Psychology of Mathematics Education,2003:Honolulu*

Notes:ToonTalk is a commercial product.Free trial and Beta versions are available fromhttp://www.ToonTalk.com.We have designed a set of tools to assist in the programming tasks.These are available on the web reports system under the Tools section.See weblabs.eu.com To read more about the WebReports system,and to see it in action,visit:http://www.WebLabs.org.uk/wlplone/Help/about_index_html.The main "Guess my rule" page can be found on http://www.WebLabs.org.uk/wlplone/Members/yish/my_reports/Report.2004-01-06.5353

Acknowledgement:We acknowledge the support of the European Union,Grant # IST-2001-32200. *WebLabs* is a three-year European research project on the use of programming and web-based collaboration in mathematics and science education.Our focus is on communities of young learners (10-14 years),engaged in collaborative modelling of mathematical and scientific phenomena,across five EU countries.